محاضرة ٤ : أساسيات Digital Image - العمليات الرياضية

المحتوى :
① array & matrix operations
② linear & non-linear operations
③ Arithmatic operations
④ Sets & Logic operations

Array & Matrix operations

* تمثيل الصورة في شكل مصفوفة Matrix هونفسه 2D-Array
* الفرق في العمليات لما هتعامل Matrix أو 2D-Array
- الفرق زي الماتلاب لما بعمل

$x = x .\hat{} x \Rightarrow$ Array

$x = x \hat{} x =$ Matrix

* في الـ Array operations بعمل العملية على كل عنصر والمقابل له
* في الـ operations بعمل عمليات الضرب والطرح وكل حاجة زي ما احنا عارفينه في المصفوفات

* مثال توضيحي

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- Matrix operation $\Rightarrow$ Multiplication

$$A \times B = \begin{bmatrix} a_{11} b_{11} + a_{12} b_{21} & a_{11} b_{12} + a_{12} b_{22} \\ a_{21} b_{11} + a_{22} b_{21} & a_{21} b_{12} + a_{22} b_{22} \end{bmatrix}$$

ضرب مصفوفات زي ما احنا عارفينه

- Array operation $\Rightarrow$ Multiplication

$$A \times B = \begin{bmatrix} a_{11} b_{11} & a_{12} b_{12} \\ a_{21} b_{21} & a_{22} b_{22} \end{bmatrix}$$

هضرب كل عنصر من A في المقابل له من B

Linear & non-Linear operations

* بنقول إن العملية Linear لو حققت شرطين (أي شرط متحققش يبقى Non-linear operation)

- Homogeneity - Additivity

لو فرضت إني عايز أعمل عملية operation ولتكن H على صورة f(x,y) عشان أطلع صورة g(x, y)

* $H[a f(x,y)] = a H[f(x,y)] \rightarrow$ homogenous

* $H[a_i f_i(x,y) + a_j f_j(x,y)] = a_i H[f_i(x,y)] + a_j H[f_j(x,y)]$

Additive

* معنى الـ Homogeneity إنك مش فارق معايا ترتيب العمليات يعني مش فارق عملت a * f(x, y) الأول وبعد كده طبقت الـ H أو عملت H * a وطبقتها كعملية واحدة على f(x,y (الناتج ثابت)

معنى الـ Additivity إني لو بعمل عملية على صورتين مثلاً، مش فارق لو جمعت الصورتين وعملت الـ H أو لو عملت الـ H عليهم واحدة واحدة وجمعت الناتج

أمثلة للتوضيح: * عملية التجميع عملية linear

$$\sum[a_j f_i(x,y) + a_j f_j(x,y)]$$

$$= \sum[a_i f_i(x,y)] + \sum[a_j f_j(x,y)] \Rightarrow \text{Additivity}$$

$$= a_i \sum[f_i(x,y)] + a_j \sum[f_j(x,y)] \Rightarrow \text{Homogeneity}$$

* عملية إيجاد الـ Max لقيم الـ pixel مش linear

$$Max\left\{(1)\begin{bmatrix}0 & 2\\ 0 & 3\end{bmatrix} + (-1)\begin{bmatrix}6 & 5\\ 4 & 7\end{bmatrix}\right\}$$

$$= Max\left\{\begin{bmatrix}0-6 & 2-5\\ 2-4 & 3-7\end{bmatrix}\right\} = Max\left\{\begin{bmatrix}-6 & -3\\ -2 & -4\end{bmatrix}\right\} = \boxed{-2}$$

But: $(1)\, Max\left\{\begin{bmatrix}0 & 2\\ 2 & 3\end{bmatrix}\right\} + (-1)\, Max\left\{\begin{bmatrix}6 & 5\\ 4 & 7\end{bmatrix}\right\}$

$= \boxed{3 - 7 = -4 \neq -2}$ [2]

So, Max is non-linear operation.

Arithmatic Operations

عمليات الحساب كلها (ضرب ، جمع ، طرح ، قسمة) بتتم ك Array operations

لكل عملية تطبيقات هنشوفها:

① Addition (الجمع)

- من التطبيقات المهمة عملية ال noise reduction .
- الفكرة إنك أحيانا بعض ال sensors بتجيب Noise مع الصورة
- هلقط كذا صورة لنفس المشهد وأعمل averaging ليهم ، هلاقي ال noise بتقل
- ال averaging هو أجمع الصور (عملية الجمع هتبقى Array operation) وبعد كده أقسم على عدد الصور (كل بكسل هقسمه على عدد الصور)
- كل ما زادت الصور بقى عندي نتيجة أحسن في تقليل ال noise
- نوع ال noise اللي بينفع معاه ال averaging أكتر هي ال gaussian noise

راجع سلايد [4,5] وصفحة 19 (بترقيم البرنامج مش الورق)

② Subtraction (الطرح)

- من التطبيقات المهمة بيبقى في الصناعة مثلا (اكتشاف العيوب ومراقبة الجودة) أو في الطب (لتقدير الأنسجة وتشوف السائل بيتحرك إزاي) أو حتى في الخرائط لتوضيح الطرق مثلا أو صور الأقمار الصناعية

* فكرة المصانع في مراقبة الجودة مثلا: بيكون عندي صورة أصل بقارن بيها ، اسمها ال Golden image ولتكن مثلا لعينة سليمة من المنتج ، كل منتج بعدي هاخدله صورة وأطرحها من ال Golden image ، لو فيه عينة فيها قطعة ناقصة مثلا هيبقى فيه قيم لل pixel في المنطقة دي والباقي هيبقى تقريبا بصفر.

* فكرة الخرايط من الأقمار الصناعية (راجع سلايد [4.6])
- هيلقط صورة، وبعد كده ياخد الصورة دي يعمل منها نسخة ويخلي الـ bit رقم صفر (least significant bit) بصفر، ويطرح الصورتين من بعض، هتوضحله المسارات أكتر.

* فكرة المجال الطبي (راجع سلايد [4.7])
- هيلقط صورة بيسميها الـ Mask
- بعدها يحقن المريض بمادة مشعة بحيث يعرف يلقط live image
- هياخد الصورة الـ live image ويطرحها من الـ Mask
- يعمل عليها شوية Enhancement عشان يقدر يشوف تحرك المادة في جسم المريض

راجع صفحة 20 في ملف السنة اللي فاتت

③ Multiplication (الضرب)
راجعه من الـ slides [4.8] و [4.9] ومن ملفات السنة اللي فاتت صفحة 20

---

* وارد جدا بعد أي عملية إن قيمة الـ pixel تعدي أعلى قيمة مسموحة أو تبقى بالسالب، في الحالة دي بعمل rescale و normalize قبل ما أحفظ الصورة، وإلا هيعملها clipping

- أولا: لو عندي قيم سالبة، هشوف أصغر قيمة سالبة وأطرحها من كل الـ Pixels وتطلع صورة $f_m(x,y)$

$$f_m(x,y) = f(x,y) - \min[f(x,y)]$$

- ثانيا، هاخد قيمة أعلى pixel عندي وأقسم كل الـ pixels عليها هتديني قيم بين صفر و واحد، أضربها بقى في أعلى intensity مسموحة عندي، وهي L اللي قلنا قبل كده $L = 2^K$

$$f_s(x,y) = L\left[\frac{f_m(x,y)}{\mathrm{Max}[f_m(x,y)]}\right]$$

هتديني صورة قيم الـ pixel فيها بين (0 , L)

مثال:

$$f_1(x,y) = \begin{bmatrix} 0 & 50 & 250 \\ 7 & 35 & 70 \\ 10 & 40 & 0 \end{bmatrix}, \quad f_2(x,y) = \begin{bmatrix} 10 & 90 & 30 \\ 0 & 80 & 70 \\ 5 & 4 & 33 \end{bmatrix}$$

$$f_3(x,y) = \begin{bmatrix} 20 & 10 & 0 \\ 30 & 255 & 70 \\ 1 & 40 & 10 \end{bmatrix} \qquad K = 8 \Rightarrow L = 256$$

$$f(x,y) = f_1(x,y) + f_2(x,y) - f_3(x,y)$$

$$= \begin{bmatrix} -10 & 130 & 280 \\ -23 & -140 & 70 \\ 14 & 4 & 23 \end{bmatrix}$$

① هاخد أصغر قيمة سالبة وهي -140 واطرحها من كل الـ Pixels

$$f_m(x,y) = \begin{bmatrix} 130 & 270 & 420 \\ 117 & 0 & 210 \\ 154 & 144 & 163 \end{bmatrix}$$

② هنفذ الخطوة الثانية على جزئية، عندي أكبر قيمة هي 420 فهقسم كله عليها

$$\frac{f_m(x,y)}{Max[f_m(x,y)]} = \begin{bmatrix} 0.3095 & 0.6429 & 1 \\ 0.2786 & 0 & 0.5 \\ 0.3667 & 0.3429 & 0.0548 \end{bmatrix}$$

هضرب في L اللي هي 256 [الصح نضرب في 255 = L-1 بس نمشي زي ما المحاضرة قالت] ونقرب الناتج لأعداد صحيحة

$$f_s(x,y) = L\left[\frac{f_m(x,y)}{Max(f_m(x,y))}\right] = \begin{bmatrix} 79 & 165 & 256 \\ 71 & 0 & 128 \\ 94 & 88 & 14 \end{bmatrix}$$

(256 ← المفروض تبقى 255)

Sets and Logic operations

* راجع سلايد [4.12] ، بيعرف فيها يعني إيه set والعمليات اللي ممكن نعملها زي Union (U) و intersection (∩) وال difference وال Complement

* سلايد [4.13] بتوضح شكل العمليات دي

* نفس الكلام في سلايد [4.14] ، سلايد [4.15] بتوضح تنفيذ العمليات دي على صور ، مكتوب على اليمين العملية اللي حصلت على كل صورة

* ال Logic operation موضحها على صورة Binary في سلايد [4.16] .

لو هنطبق ال Logic operation على صورة K = 8 مثلا ، هتبقى العمليات Bitwise operation (هتشتغل على كل pixel على مستوى كل bit)

* سلايد [4.17] مش مهمة قوي ، المفروض عارف الكلام اللي فيها ، اقرأها وبس